**COMPARISON AND EVALUATION OF DEEP LEARNING MODELS FOR CLASSIFICATION OF SPUTUM SMEAR IMAGES**

**INT 400 – INTERNSHIP 3 PROJECT REPORT**

***Submitted by***

**NIVEDITHA B - E0119019**

***In partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**(Artificial Intelligence & Machine Learning)**

**Sri Ramachandra Faculty of Engineering and Technology**

**Sri Ramachandra Institute of Higher Education and Research, Porur, Chennai -600116**

**DECEMBER, 2021**

**COMPARISON AND EVALUATION OF DEEP LEARNING MODELS FOR CLASSIFICATION OF SPUTUM SMEAR IMAGES**

**INT 400 – INTERNSHIP 3 PROJECT REPORT**

***Submitted by***

**NIVEDITHA B, E0119019**

***In partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**(Artificial Intelligence & Machine Learning)**

**Sri Ramachandra Faculty of Engineering and Technology**

**Sri Ramachandra Institute of Higher Education and Research, Porur, Chennai -600116**

**DECEMBER, 2021**

**BONAFIDE CERTIFICATE**

Certified that this project report **“COMPARISON AND EVALUATION OF DEEP LEARNING MODELS FOR CLASSIFICATION OF SPUTUM SMEAR IMAGES”** is the bonafide record of work done by **“NIVEDITHA B – E0119019”** who carried out the internship work under my supervision.

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature of Faculty Member**   |  | | --- | | **Dr. Vanitha V**  **Assistant Professor,**  Department of Computer Science and Engineering  Sri Ramachandra Faculty of Engineering and Technology,  SRIHER, Porur, Chennai-600 116. | | **Signature of Faculty Member**   |  | | --- | | **Dr. Pitchumani Angayarkanni**  **Associate Professor,**  Department of Computer Science and Engineering  Sri Ramachandra Faculty of Engineering and Technology,  SRIHER, Porur, Chennai-600 116. | |
| **Signature of Faculty Member**   |  | | --- | | **Prof. Nirmala B**  **Assistant Professor,**  Department of Computer Science and Engineering  Sri Ramachandra Faculty of Engineering and Technology,  SRIHER, Porur, Chennai-600 116. | | **Signature of Vice-Principal**   |  | | --- | | **Prof. M. Prema**  **Vice-Principal,**  Department of Computer Science and Engineering  Sri Ramachandra Faculty of Engineering and Technology,  SRIHER, Porur, Chennai-600 116. | |

**Evaluation Date:** 29th Jan 2022

**ACKNOWLEDGEMENT**

I express my sincere gratitude to our Chancellor, Vice-Chancellor and our sincere gratitude to our Provost **DR. RAJU** and our Vice-Principal **PROF. PREMA M** for their support and for providing the required facilities for carrying out this study.

I wish to thank my faculty supervisor(s), **DR. VANITHA V, DR. PITCHUMANI ANGAYARKANNI, PROF. NIRMALA B,** Department of Computer Science and Engineering, Sri Ramachandra faculty of Engineering and Technology for extending help and encouragement throughout the project. Without their continuous guidance and persistent help, this project would not have been a success for me.

I am grateful to all the Sri Ramachandra Faculty of Engineering and Technology members, my beloved parents and friends for extending the support, who helped us overcome obstacles in the study.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | **7** |
|  | **LIST OF FIGURES** | **8** |
|  | **LIST OF ABBREVIATIONS** | **9** |
|  | **LIST OF TABLES** | **9** |
| **1** | **INTRODUCTION** | **10** |
|  | 1.1 Introduction | 10 |
|  | * 1. Problem Statement | 10 |
|  | 1.3 Motivation | 10 |
|  | 1.4 Objective | 11 |
| **2** | **LITERATURE SURVEY** | **12** |
| **3** | **METHODOLOGY AND WORKFLOW** | **14** |
|  | 3.1 Methodology | 14 |
|  | 3.2 Workflow | 15 |
| **4** | **IMPLEMENTATION** | **16** |
|  | 4.1 Data Collection and Generation | 16 |
|  | 4.2 Data Pre-Processing | 18 |
|  | 4.3 Train and Test Datasets | 18 |
|  | 4.4 Models Used | 19 |
|  | 4.4.1 Inceptions - V3 | 19 |
|  | 4.4.2 VGG 16 | 20 |
|  | 4.4.3 ResNet - 50 | 21 |
| **5** | **RESULTS AND DISCUSSIONS** | **23** |
|  | 5.1 Model’s Predictions | 23 |
|  | 5.2 Evaluating Models | 24 |
|  | 5.2.1 Inception – V3 | 24 |
|  | 5.2.2 ResNet - 50 | 25 |
|  | 5.3 Comparing Models | 26 |
| **6** | **CONCLUSION** | **27** |
|  | **REFERENCES** | **28** |
|  | **WORKLOG** | **31** |
|  | **INTERNSHIP OFFER LETTER** | **34** |
|  | **CERTIFICATE OF COMPLETION** | **35** |

**ABSTRACT**

Tuberculosis is a bacterial infection that spreads in addition to any normal shape of bloodless or flu, even though it is not as contagious as them. When one receives TB, it affects the lungs and that might be substantially seen. But it is able to furthermore have an effect on any part of the frame together with the abdomen, glands, bones, and anxious system. That also can moreover rise up first-rate in a probably important condition. However, it is able to be cured if it is detected early and treated with antibiotics. It might also additionally take numerous weeks even earlier than the inflamed character notices that they'll be unwell. About one place of the world’s populace has a TB germ into the air. In 2020, extra human beings died from TB, with manner fewer human beings being recognized and treated or provided with TB preventive remedy in assessment with 2019, and regular spending on essential TB offerings falling. WHO estimates that a few 4.1 million human beings presently be troubled thru TB however have no longer been recognized with the sickness or have no longer formally endorsed to countrywide authorities. Sputum smear microscopy is the sign used device for MTB prognosis in maximum growing international locations due to the fact it is a lousy lot ton much less costly. Manual detection of bacilli from stained sputum images is time-ingesting due to the fact it can take 15 mins in keeping with slide for detection, lowering the huge style of slides which affects the accuracy of the output. Thus, computer-aided automatic techniques offer manifestly a most fantastic answer in sickness prognosis internal a lousy lot tons much less time and without fairly expert laboratory experts. Automatic techniques are typically taken into consideration as an opportunity for this problem. Attempts have been made to make bigger automatic techniques to discover TB microorganisms from microscopic sputum smear images. This document might offer a top-degree view of automatic techniques primarily based totally on image processing techniques. The document shows the evaluation of 3 deep learning models, InceptionV3, VGG16, Resnet50, which are used to search ahead to detect the disease using the sputum smear images.

*Keywords: Tuberculosis, Sputum smear microscopic images, Comparison of classification models.*

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| 3.1 | Research Methodology | 14 |
| 3.2 | Research Process | 15 |
| 3.3 | Workflow | 15 |
| 4.1 | Image data from Kaggle | 16 |
| 4.2 | Sample TB positive images from Kaggle | 17 |
| 4.3 | TB negative sputum smear image   that was given for augmentation | 17 |
| 4.4 | Sample of the 60 Augmented TB negative sputum smear images | 17 |
| 4.5 | Labeling Images | 18 |
| 4.6 | Convert labels to NumPy arrays | 18 |
| 4.7 | Labeling images | 18 |
| 4.8 | Train and test dataset | 19 |
| 4.9 | Building Inception V3 | 19 |
| 4.10 | Training Inception V3 | 20 |
| 4.11 | Building RestNet50 | 21 |
| 4.12 | Training ResNet | 22 |
| 5.1 | Model Prediction Sample 1 | 23 |
| 5.2 | Model Prediction Sample 2 | 23 |
| 5.3 | Inception V3 model accuracy | 24 |
| 5.4 | Inception V3 model evaluation metrics | 24 |
| 5.5 | Inception V3 confusion matrix | 24 |
| 5.6 | ResNet50 loss plot | 25 |
| 5.7 | ResNet50 model accuracy | 25 |
| 5.8 | ResNet50 evaluation metrics | 25 |
| 5.9 | ResNet50 confusion matrix | 25 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATIONS** | **EXPANSION** |
| **TB** | Tuberculosis |
| **MTB** | Mycobacterium tuberculosis |
| **ML** | Machine Learning |
| **DL** | Deep Learning |
| **NN** | Neural Network |
| **CNN** | Convolutional Neural Network |
| **ResNets** | Residual Networks |
| **Conv** | Convolutional |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO.** | **TITLE** | **PAGE NO.** |
| 5.3.1 | Performance of Models | 26 |

**CHAPTER – 1**

**INTRODUCTION**

* 1. **Introduction**

Tuberculosis is caused by Mycobacterium tuberculosis, a slow-growing bacteria that thrive in areas of the body that are rich in blood and oxygen, such as the lungs.

It spreads through infected droplets, released in the air by coughing, sneezing, etc, by the affected individual. It usually spreads after prolonged exposure to the infected individual. Immunosuppressed (with weak immunity) individuals are at higher risk of contracting the infection.

* 1. **Problem Definition**

To decrease the faster spread of TB, it is essential to detect that infection in people faster. The detection method should be easily accessible by people and also be time-efficient. Since the easiest way to detect it is from sputum smear tests, which are prone to error and time-consuming if done manually. So while seeking the help of ML, the forthcoming step would be classification. Classification of medical data is an important task in the prediction of any disease. Various ML models can be used to classify any given microscopic sputum smear image into TB positive or negative class (binary classification). The intention of this initiative is to analyze and compare deep learning models that could help classify those sputum smear images automatically.

* 1. **Motivation**

As said earlier, TB is due to a pathogenic microorganism called Mycobacteria. It is airborne and may have an impact on personal elements which consist of the lungs, brain, and spine. In 2020, its miles decided to be the 13th major motive of dying throughout the world. Fortunately, it's miles treatable at the same time as decided earlier stages. United Nations Sustainable Development Goals (SDGs) have set a motive to forestall the TB epidemic through manner of approach of 2030. There are numerous strategies to discover TB, out of which sputum smear microscopy is considered to be correct. However, it is not considered to be an inexperienced way as it's miles time-ingesting and additional liable to errors, thinking about the reality that it's miles done manually. Instead, ML can help in detecting it rapidly with greater accuracy and fewer errors. ML practitioners have been working to create such models. So, our motive is to assemble a deep studying model that is capable of stumbling on and discerning out MTB, it's the fundamental reason of Tuberculosis thru microscopic images, and thereby make quicker and additional correct diagnoses and thereby make faster and more accurate diagnoses.

* 1. **Objectives**
* To understand various image processing techniques.
* To perform augmentation and other image processing techniques.
* To train CNN models on TB sputum smear images and do classification.
* To evaluate and compare the models.

**CHAPTER - 2**

**LITERATURE SURVEY**

TB can be diagnosed using several different methods. Among them, the two ways which use microscopy are the fluorescent method and the other one is the conventional method.

***Fluorescent Method,*** Veropuolos et al.[05](1998) presented a model, which used Fourier features and Edge detection, morphological operators for segmentation, was able to score 97.9% accuracy, 94.1 % •sensitivity, 99.1 % specificity. Forero-Vargas et al. [06] (2001) experimented with fuzzy segmentation using color information. But later in 2002, he modified it to fuzzy segmentation based on the chromatic histogram, where he used top-hat operation for bacilli detection and fisher linear discriminant for classification. That model got 82% specificity. When the Heuristic knowledge-based approach for bacilli detection was followed by Forero MG et al. [07] (2003), it gave them ~98% sensitivity. In 2004, they used shape-based heuristic knowledge for classification tree construction, which gave them a specificity between 87–99 %, depending upon the distance threshold. Later in 2006, the use of a Bayesian classifier for classification and morphological operations preceded by edge detection for segmentation rendered a sensitivity of ~95%. The first large-scale evaluation of an automated microscopy system for tuberculosis (TBDx) in 2012 by Lewis et al. [08], gave 75.8 % sensitivity and 43.5 % specificity. Priya et al.[09] used some statistical features such as mean, skewness, and kurtosis, and the DE-ELM method is used for classification to commit a model with 100 % specificity and 92.5 % accuracy. When a set of rotation and translation invariant features like object major and minor axis length, area, Fourier descriptors, etc. were devoted in a model by Santiago-Mozos et al. [10] (2014), it hold out to73.53 % sensitivity and 99.99 % specificity. Implementation of LED fluorescent microscopy by Jager et al. [11] reached 83.97 % sensitivity and 85.56 % specificity.

***Conventional Method,*** Using sputum smear microscopy for TB diagnosis is the conventional method. A debut to this method was made by Costa et al.[12] (2008) which used morphological filters and adaptive thresholding. It achieved 76.65% sensitivity but the false-positive rate was 12%. Bayesian and Hue-based segmentations were also approached. In 2009, a model was raised by Sotaquira et al. [13], which can identify the level of infection based on the number of bacilli’s/field. Segmentation using thresholding based on the first derivative helped them to achieve 90.9% sensitivity, ~100% specificity with 85.7% accuracy. A success rate of 93.5% was achieved by Nayak et al.[14] (2010), whose model was able to count the number of AFB even if they had beaded structure. In the same year, Zhai et al.[15] found a faster method that can detect bacilli within 3 –5 mins. They used a Decision Tree classifier which worked up to 100% sensitivity but accuracy was less than 80%. Another breakthrough in that year was by Khutlang et al.[16] , who considered Fourier features, Moment features, eccentricity, the value of the central pixel and performed bacilli segmentation using a combination of two class pixels classifier to achieved 98.55% accuracy with 97.77 % sensitivity and 99.13 % specificity. Simultaneously, Osman et al.[17] got an accuracy of 86.32% by using a genetic algorithm-based neural network, 99.82% by using modified recursive prediction error (MRPE) for training HMLP, 77.25% with ELM algorithm, and 75.46% by using C-SLFN trained by improved ELM algorithm. Every one of his methods used K-means for segmentation. CostaFilho et al. [18] (2012) used color ratio as an additional feature, which lead to 91.49% precision, 91.53 % sensitivity, and a false detection rate was 8.51 %. Elsevier [19](2018) published a model which used deep learning methods and performed detection of TB, by image binarization and subsequent classification of detected regions using a convolutional neural network. That algorithm was evaluated using a dataset of 22 sputum smear microscopic images with different backgrounds (high density and low-density images) and achieved 97.13% recall, 78.4% precision, and 86.76% F-score. In that model images were binarized using Otsu's method after converting to grayscale but when the object area is small compared to the background area Otsu fails. The Journal of Infection in Developing Countries (2020) produced a transfer learning model which used a pre-trained AlexNet to detect microscopic stain images that contain Mycobacterium tuberculosis bacilli to make accurate clinical decisions. The model achieved 98.15 % accuracy, 96.77% sensitivity, and 100% specificity. Dataset was collected from two different centres which use different microscopic magnification ranges and backgrounds.

**CHAPTER – 3**

**METHODOLOGY & WORKFLOW**

**3.1 METHODOLOGY: RESEARCH METHODOLOGY**

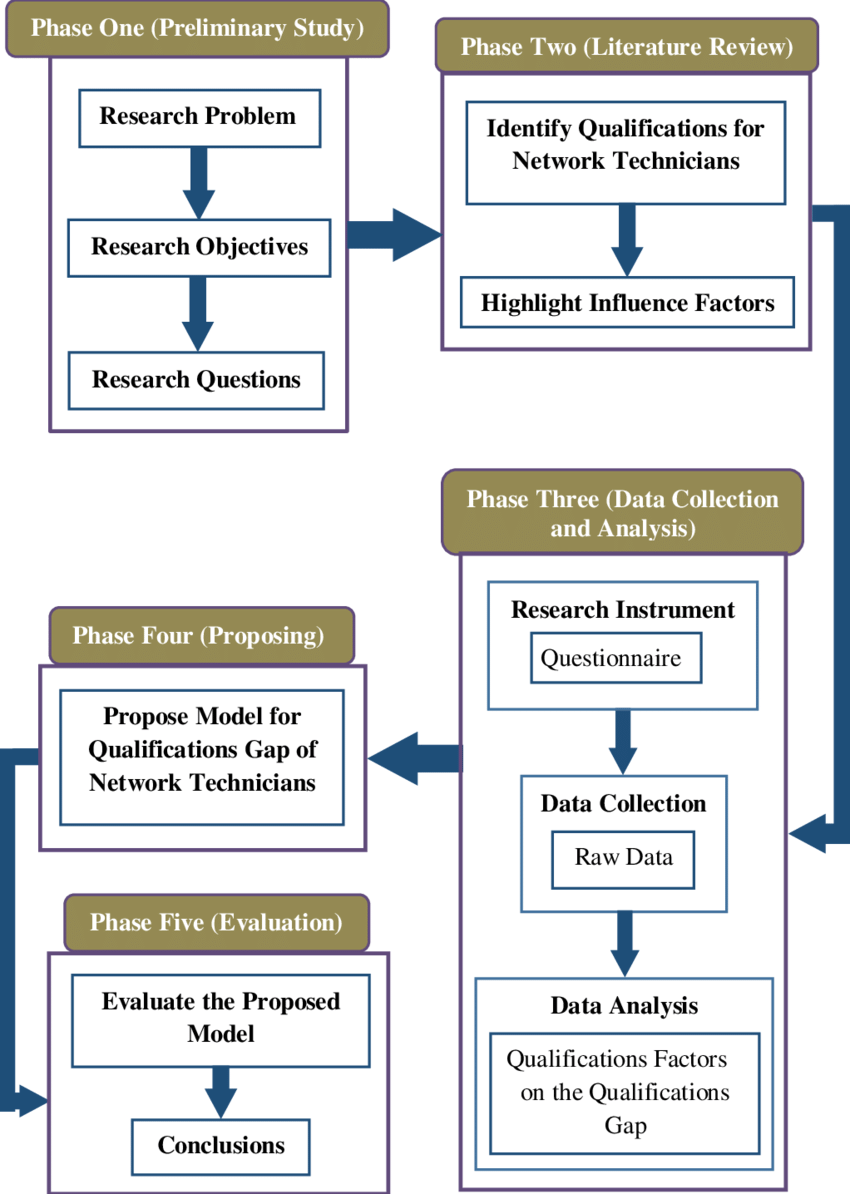


Fig-3.1 Details the working flow the

Research Methodology , which was followed along the

project period.



Fig-3.2 Portraits the progressive steps followed along

the research process.

**3.2 WORKFLOW**

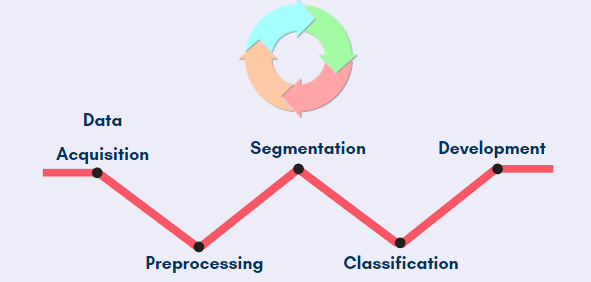


Fig-3.3 Depicts the procedural steps of the project workflow.

**CHAPTER - 4**

**IMPLEMENTATION**

**4.1 DATA COLLECTION AND GENERATION**

The dataset of positive TB images available in the Kaggle, public source platform has been used here. It has 1253 sputum smear images. Since only positive images were available, a single negative image was passed to the Augmentor pipeline. A series of augmentation functions were set to be applied, to the newly generated images. Any newly generating images were given a 50% probability of random brightness, 70% probability of random contrast, 70% probability to rotate, and 50% probability to rotate90, rotate270, flip-right-left, flip-top-bottom. 60 more new negative images with varied lighting, brightness, and contrast were generated by this process. Thus we can eliminate the class imbalance problem.

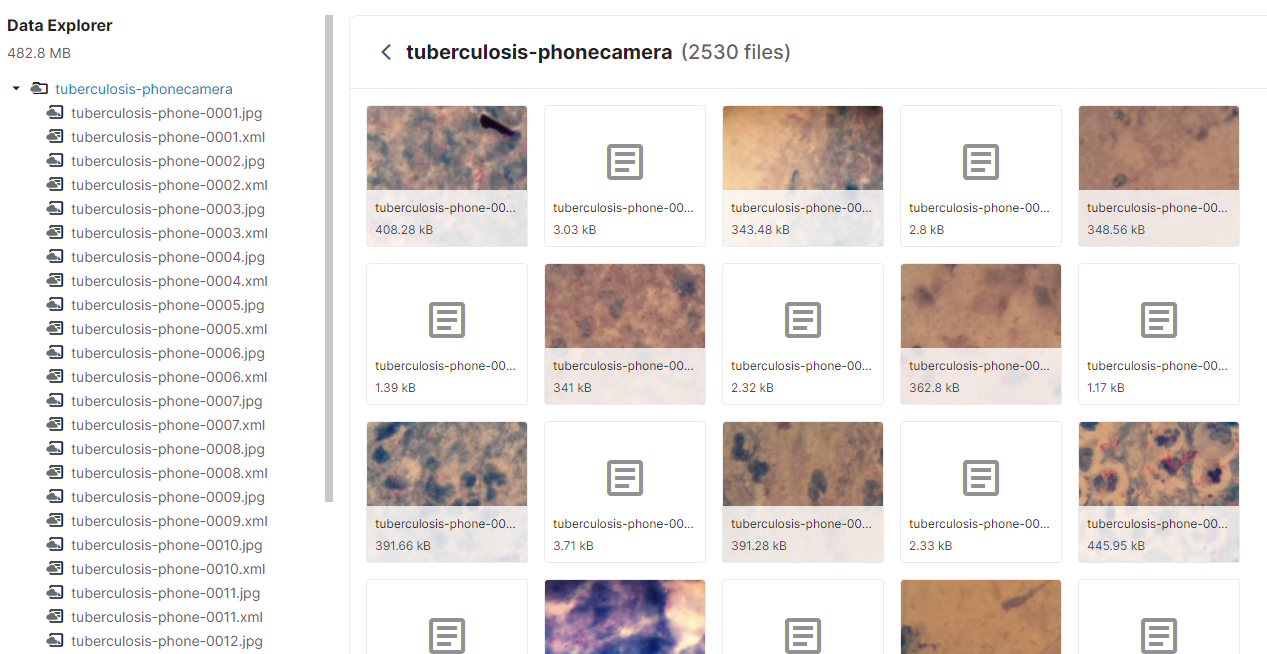
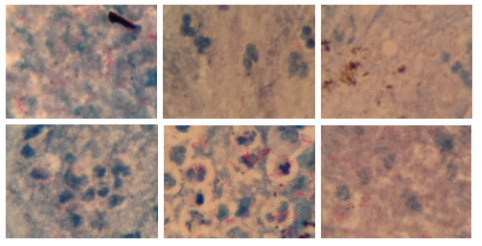


Fig-4.1 Image data from Kaggle public repository. Gives overview of the sputum smear image data that were collect from various sources. There is also the corresponding XML files of the image which describes the position of the TB bascilli in the sputum smear images.

  
Fig-4.2 Sample of some the TB positive images that were

extracted from Kaggle open repository.

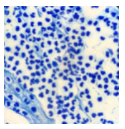


Fig-4.3 TB negative sputum smear image

 that was taken to perform augmentation

 to generate more images.

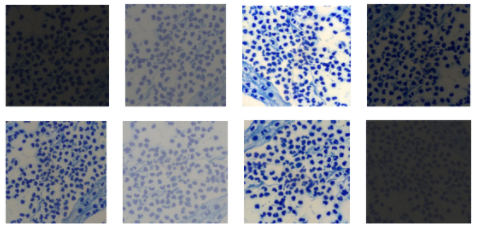


Fig-4.4 Picture of some of the 60

Augmented TB negative sputum smear images, that were

Generated from the single negative image in Fig-4.3.

**4.2 DATA PRE-PROCESSING**

**4.2.1 Labelling Images**

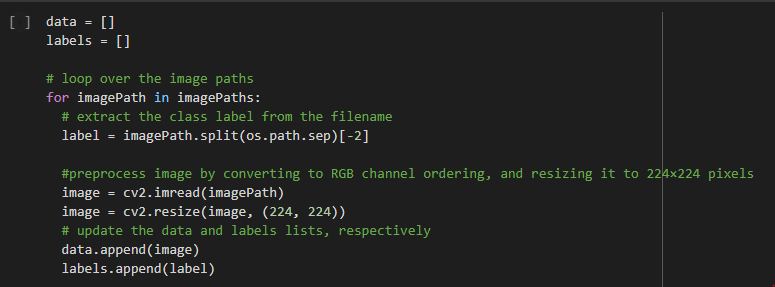


Fig-4.5 Snippet of code that was used to

Label the TB sputum smear images that were collected and generated.

They are labels as either TB positive or TB negative.

**4.2.2 Converting Labels to NumPy arrays**

Convert the data and labels to NumPy arrays while scaling the pixel intensities to the range [0, 1]

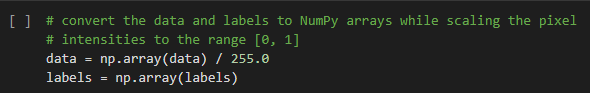


Fig-4.6 Snippet of code that was used to

Convert labels to NumPy arrays in order to use them

as tensors to feed the CNN model.

**4.2.3 Encoding Labels**

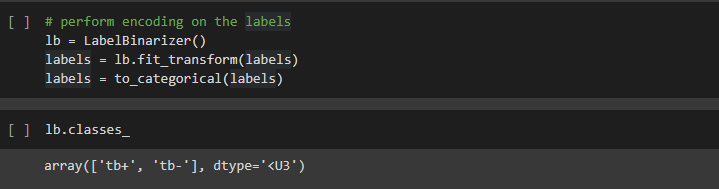


Fig-4.7 Labelling images using LabelBinarizer

Function in Python.

**4.3 TRAIN AND TEST DATASET**

80% of data for training and 20% of data for testing.

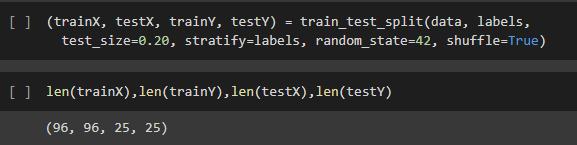


Fig-4.8 Snippet of code that was used to split the whole images dataset

Into Train and test set. Then, the length of the train and test input as well

as the target variables found here.

**4.4 MODELS USED**

**4.4.1 InceptionV3**

In InceptionV3, essentially a few distinctive length filters are used parallelly to seize the records in a photograph. Then, factorization of these filters is accomplished to lessen the number of parameters. That manner could lessen the number of computational assets used.

The dense layer classifier earlier than the SoftMax classifier withinside the structure is likewise referred to as Auxiliary Classifiers. The Loss of that Auxiliary Classifier is used to triumph over Vanishing Gradient hassle. To enhance the results, that loss is introduced to the primary category loss. Auxiliary Classifiers do not regulate the prediction. The SoftMax Classifier is the only which does the Predictions.

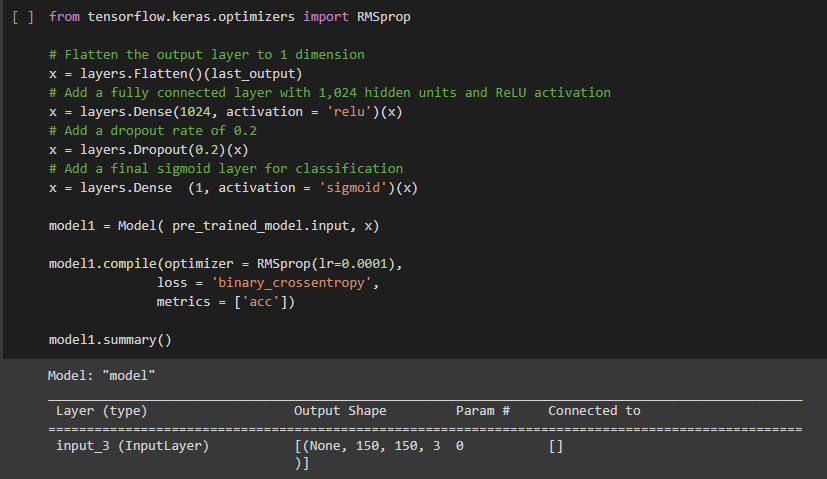


Fig-4.9 Snippet of code that was used to

Building Inception V3 CNN. After building, the model summary

is displayed.

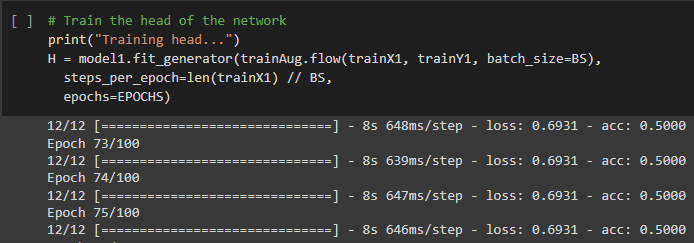


Fig-4.10 Snippet of code used to

Training that Inception V3 CNN that built as shown in Fig-4.09.

**4.4.2 VGG16**

VGG16 is a convolutional neural community version. The version achieves 92.7% top-five take a look at accuracy in ImageNet, that's a dataset of over 14 million photos belonging to one thousand classes. It turned into one of the well-known versions. It makes the development over AlexNet through changing massive kernel-sized filters (eleven and five withinside the first and 2d convolutional layer, respectively) with a couple of 3x3kernel-sized filters one after another.

The input to cov1 layer is of constant length 224 x 224 RGB photographs. The photograph is handed thru a stack of convolutional (Conv.) layers, wherein the filters have been used with a totally small receptive field: 3x3(that's the smallest length to seize the perception of left/right, up/down, center). In one of the configurations, it additionally makes use of 1×1 convolution filters, which may be visible as a linear transformation of the enter channels (accompanied through non-linearity). The convolution stride is constant to one pixel; the spatial padding of Conv. layer enter is such that the spatial decision is preserved after convolution, i.e. the padding is 1-pixel for 3×3 Conv. layers. Spatial pooling is achieved through 5 max-pooling layers, which observe a number of the Conv. layers (now no longer all of the conv. layers are accompanied through max-pooling). Max-pooling is completed over a 2×2 pixel window, with stride 2.

The sputum smear photos dataset that turned into taken from the kaggle has been used to educate this version, in conjunction with the pattern of 2 hundred augmented photos of TB photograph. The photograph statistics have been transformed into the NumPy supply document wherein the records of the pixels of the photos are shops and those NumPy data files have been used to educate the version after splitting the dataset into educating and take a look at.

**4.4.3 ResNet50**

RestNet50 is a Residual Network with 50 layers. The most important precept on which ResNets works is to construct a deeper community, in comparison to different parallel networks and concurrently discover an optimized quantity of layers to negate the vanishing gradient hassle. Because, in Deep Neural Networks which has a massive quantity of layers, there usually is the hassle of Vanishing Gradient, which gets up because of the chain rule accompanied whilst lower backpropagation. But, ResNets triumph over this hassle through the use of an idea series Skip Connection, wherein the authentic entry of a layer is introduced to the output of that precise layer itself. In this method, the output can be the same as the sum of features turn of entering + enter. So, in ResNets the weights must be shifted in accord to make the Function of entering to 0, to lessen the loss.

The entered photograph needs to be of length 224\*224 for RestNet50. Ninety-seven photos have been handed to the educating the version for 100epochs. This version led to 92% accuracy for the take a look at statistics which includes 24 photos.

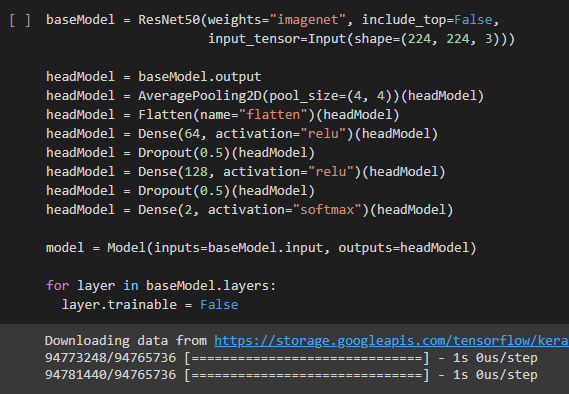


Fig-4.11 Snippet of code that was describes

The building of RestNet50 CNN.

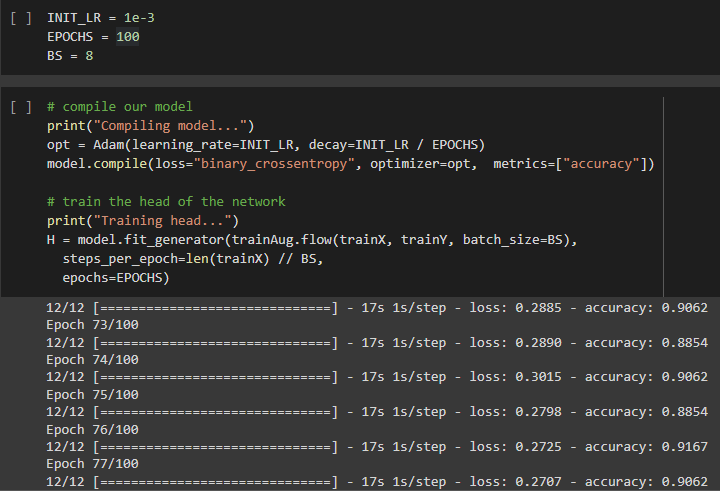


Fig-4.12 Snippet of code that was used

To training the ResNet50 CNN that was built as shown in Fig-4.11.

**CHAPTER - 5**

**RESULTS AND DISCUSSION**

**5.1 MODEL PREDICTIONS :**

A function named ‘resnet50\_predict’ was defined to predict a sputum smear image class.

Sample 1

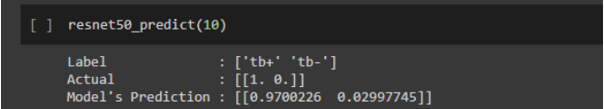


Fig-5.1 Model Prediction Sample 1 – The 10th image

From the test data is passed to a defined predict function.

The actual as well as predicted class along with actual labels is displayed as result.

A positive TB image from test data was passed into the function. The model also predicts that it is a positive TB image with 97% confidence.

Sample 2

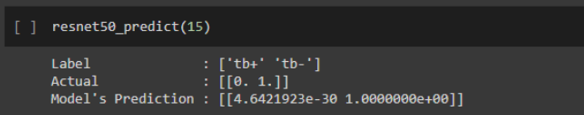


Fig-5.2 Model Prediction Sample 2 - The 15th image

From the test data is passed to a defined predict function.

The actual as well as predicted class along with actual labels is displayed as result.

Then, a negative TB image from test data was passed into the function. The model is 100% confident that it is a TB negative image.

**5.2 EVALUATING MODELS**

**5.2.1 Inception V3**

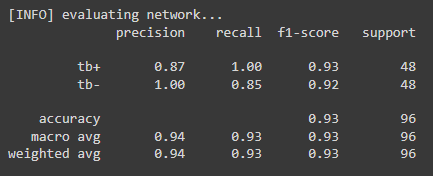


Fig-5.3 Describes the Inception V3 model’s

Precision, recall, f1-score, accuracy and weighter average of

Tb positive and tb negative class.



Fig-5.4 Inception V3 model evaluation metrics is calculated

In terms of accuracy, Sensitivity and Specificity the determine True Positive

And True Negative prediction rate.

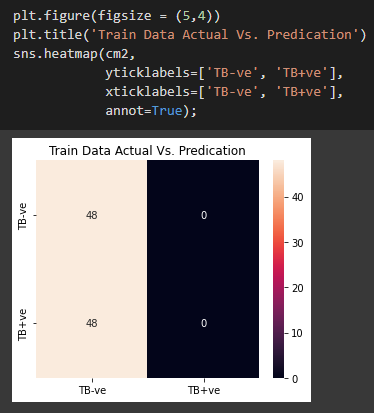


Fig-5.5 Inception V3 confusion matrix displays

The True positive, True Negative, False positive and False Negative rate

Of the model’s predictions.

**5.2.2 ResNet50**

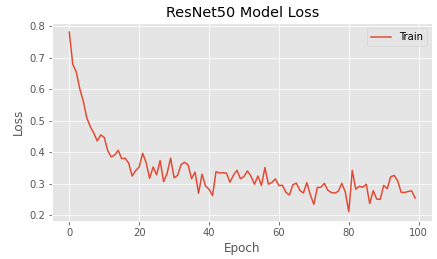


Fig-5.6 ResNet50 loss plot – The loss

Is higher at the starting epoch. But it gradually decreases

As the number of the epochs increases. The loss is comparatively

Reduced at the end of 100th epoch.

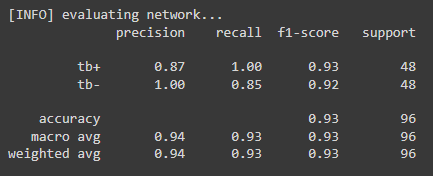


Fig-5. 7 Describes ResNet50 model’s

Precision, recall, f1-score, accuracy and weighter average of

Tb positive and tb negative class.



Fig-5.8 ResNet50 evaluation metrics is calculated

In terms of accuracy, Sensitivity and Specificity the determine True Positive

And True Negative prediction rate.

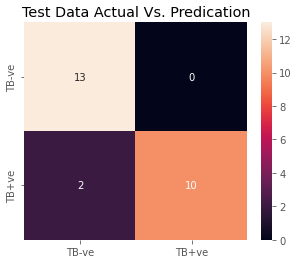


Fig-5.9 ResNet50 confusion matrix displays

The True positive, True Negative, False positive and False Negative rate

Of the model’s predictions.

**5.3 COMPARING PERFORMANCE OF MODELS**

Table-5.3.1 Performance of Models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model No.** | **Model** | **Accuracy** | **Sensitivity** | **Specificity** |
| 1 | InceptionV3 | 0.52 | 1.00 | 0.00 |
| 2 | VGG16 | 0.83 | 0.76 | 0.89. |
| 3 | ResNet50 | 0.9271 | 1.0000 | 0.8542 |

**CHAPTER – 6**

**CONCLUSION**

Worked on CNN models like Residual Networks, VGG, and Inception to perform classification on TB sputum smear medical images. Performed augmentation and other image processing techniques on those images before passing them to the model. The evaluation was done on trained models. In comparison, we can infer that ResNet-50 performs better than VGG16 and Inception-V3. Inception and VGG are created to serve the purpose of reducing the computational burden of deep neural nets while, ResNet focuses on computational accuracy, which makes it perform better. Hence, ResNet could evidently show better performance compared to other models.

**CHAPTER – 7**

**REFERENCES**

[01] ResNet50 Architecture : <https://www.youtube.com/watch?v=Uuc1wdqMFtQ>

[02] Train NN: <https://www.youtube.com/watch?v=1Gbcp66yYX4>

[03] Inception V3 Architecture: <https://www.youtube.com/watch?v=ekixid21Q5U&t=0s>

[04] Literature Survey : <https://mail.google.com/mail/u/2/#search/paper+for+reference/FMfcgzGlksFnRqHgmwdMPlRmwNmWcHQg?projector=1&messagePartId=0.1>

<https://mail.google.com/mail/u/2/#search/paper+for+reference/FMfcgzGlksFnRqHgmwdMPlRmwNmWcHQg?projector=1&messagePartId=0.2>

<https://mail.google.com/mail/u/2/#search/paper+for+reference/FMfcgzGlksFnRqHgmwdMPlRmwNmWcHQg?projector=1&messagePartId=0.3>

<https://mail.google.com/mail/u/2/#search/paper+for+reference/FMfcgzGlksFnRqHgmwdMPlRmwNmWcHQg?projector=1&messagePartId=0.3>

[05] Veropoulos, K., Campbell, C., and Learmonth, G., Image processing and neural computing used in the diagnosis of tuberculosis. In Proc. of IEE Colloquium on Intelligent Methods in Healthcare and 17 Page 12 of 13 J Med Syst (2016) 40:17 Medical Applications (Digest No. 1998/514), York, 8/1–8/4, 1998. doi:10.1049/ic:19981039.

[06] Forero-Vargas, M.G., Sierra-Ballen, E.L., Alvarez-Borrego, J., Pech-Pacheco, J.L., Cristobal-Perez, G., Alcala, L., and Desco, M., Automatic sputum color image segmentation for tuberculosis diagnosis. In Proc. of SPIE Algorithms and Systems for Optical Information Processing V. 4471:251–261, 2001. doi:10.1117/12. 449343.

[07]  Forero, M.G., Cristóbal, G., and Alvarez-Borrego, J., Automatic identification techniques of tuberculosis bacteria. In Proc. of Applications of Digital Image Processing XXVI (SPIE). 5203:71– 81, 2003. doi:10.1117/12.506800.

[08] Lewis, J. J., Chihota, V. N., Meulen, M., Fourie, P. B., Fielding, K. L., Grant, A. D., Dorman, S. E., Churchyard, G. J., BProof-OfConcept^ Evaluation of an Automated Sputum Smear Microscopy System for Tuberculosis Diagnosis. PLoS One, 7:11, 2012. doi:10.1371/journal.pone.0050173.

[09] Priya, E., and Srinivasan, S., Automated decision support system for tuberculosis digital images using evolutionary learning machines. European Journal for Biomedical Informatics. 9(2):3–8, 2013.

[10] Santiago-Mozos, R., Pérez-Cruz, F., Madden, M.G., and ArtésRodríguez, A., An Automated Screening System for Tuberculosis. IEEE journal of Biomedical and Health informatics. 18(3):855–862, 2014. doi:10.1109/JBHI.2013.2282874.

[11] de Jager, K., Fickling, S., Krishnan, S., Jabbari, M., Learmonth, G.W., and Douglas, T.S., Automated fluorescence microscope for tuberculosis detection. Journal of Medical Devices. 8(3):0309431– 0309432, 2014. doi:10.1115/1.4027111.

[12] Costa, M. G., Costa Filho, C. F. F., Sena, J. F., Salem, J., and Lima, M. O., Automatic identification of Mycobacterium tuberculosis with conventional light microscopy. In Proc. of 30th Annual International IEEE Eng Med Biol Soc. Vancouver, Canada, pp. 382–385, 2008. doi:10.1109/IEMBS.2008.4649170

[13] Sotaquirá, M., Rueda, L., and Narvaez, R., Detection and quantification of bacilli and clusters present in sputum smear samples: a novel algorithm for pulmonary tuberculosis diagnosis. In Proc.of International Conference on Digital Image Processing, Bangkok, 117–121, 2009.doi:10.1109/ICDIP.2009.59.

[14] Nayak, R., Shenoy, V. P., and Galigekere, R. R., A new algorithm for automatic assessment of the degree of TB-infection using images of ZN-stained sputum smear. In Proc. of International Conference on Systems in Medicine and Biology (ICSMB), Kharagpur, 294–299, 2010. doi:10.1109/ICSMB.2010.5735390.

[15] Zhai, Y., Liu, Y., Zhou, D., and Liu, S., Automatic Identification of Mycobacterium tuberculosis from ZN-stained sputum smear: Algorithm and system design. In Proc. of IEEE International Conference on Robotics and Biomimetics (ROBIO), Tianjin, 41– 46, 2010. doi:10.1109/ROBIO.2010.5723300.

[16] Khutlang, R., Krishnan, S., Dendere, R., Whitelaw, A., Veropoulos, K., Learmonth, G., and Douglas, T.S., Classification of Mycobacterium tuberculosis in Images of ZNStained Sputum Smears. IEEE Trans InfTechnol Biomed. 14(4):949–957, 2010.

[17] Osman, M.K., Ahmad, F., Saad, Z., Mashor, M. Y., and Jaafar, H., A genetic algorithm-neural network approach for Mycobacterium tuberculosis detection in Ziehl-Neelsen stained tissue slide images. In Proc. of 10th International Conference on Intelligent Systems Design and Applications (ISDA) 1229–1234, 2010. doi:10.1109/ ISDA.2010.5687018.

[18] CostaFilho, C.F.F., Levy, P.C., Xavier, C.M., Costa M.G.F., Fujimoto, L.B.M., and Salem, J., Mycobacterium tuberculosis Recognition with Conventional Microscopy. In Proc. IEEE Eng Med Biol Soc. 6263–8, 2012. doi:10.1109/EMBC.2012.6347426.

[19] Rani Oomman Panicker, Kaushik S. Kalmady, Jeny Rajan , M.K. Sabu, Automatic detection of tuberculosis bacilli from microscopic sputum smear images using deep learning methods, published in Elsevier(2018. doi: https://doi.org/10.1016/j.bbe.2018.05.007.

[20] Augmentation: <https://augmentor.readthedocs.io/en/master/_modules/Augmentor/Pipeline.html>

[21] Dropout: <https://www.youtube.com/watch?v=DNVTeGOMbLE&t=445s>

<https://www.machinecurve.com/index.php/2019/12/16/what-is-dropout-reduce-overfitting-in-your-neural-networks/>

[22] Batch Normalization: <https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/>

[23] Read XML : <https://stackabuse.com/reading-and-writing-xml-files-in-python/>

[24] Mask :

<https://www.freedomvc.com/index.php/2021/06/26/contours-and-bounding-boxes/>

<https://matplotlib.org/stable/tutorials/introductory/images.html>

<https://pysource.com/2019/02/15/detecting-colors-hsv-color-space-opencv-with-python/>

[Image Masking using OpenCV l Mask Image | Computer Vision | Data Magic - YouTube](https://www.youtube.com/watch?v=2KknXPT5CzA)

[25] Change Color of a pixel in an image: <https://www.tutorialexample.com/python-pillow-change-a-specific-color-of-an-image-a-step-guide-python-pillow-tutorialprel/#:~:text=%20Python%20Pillow%20Change%20a%20Specific%20Color%20of,of%20a%20pixel%2C%20we%20can%20use...%20More%20>

[26] Iterate through each pixel in an Image: <https://www.thecodingcouple.com/counting-pixels-by-color-in-python-with-pillow-a-pil-fork/#:~:text=Given%20a%20valid%20image%20file%2C%20the%20Python%20script,times%20the%20color%20was%20present%20in%20the%20image>.

**WORKLOG**

|  |  |  |
| --- | --- | --- |
| **Day** | **Date** | **Task Done** |
| Day 1 | 08/11/2021 | Learnt the concepts of Imade Filtering, Layer Padding, Image Thresholding, Connected Components |
| Day 2 | 9/11/2021 | DL with PyTorch (MNIST Handwritten dataset) |
| Day 3 | 10/11/2021 | Feedforward Neural Networks & GPUs |
| Day 4 | 11/11/2021 | DIP Modalities & Processing |
| Day 5 | 12/11/2021 | 1. DIP Modalities & Processing, 2. Feedforward Neural Networks |
| Day 6 | 13/11/2021 | 1.Go through papers on TB Detection, 2. DIP Modalities & Processing |
| Day 7 | 15/11/2021 | 1. Exploratory Data Analysis, 2. DIP Modalities & Processing |
| Day 8 | 16/11/2021 | 1. DL with CIFAR10 dataset |
| Day 9 | 17/11/2021 | 1. Image Classification with Convolutional Neural Networks, 2.Review 1 preparation |
| Day 10 | 18/11/2021 | 1. Image Classification with Convolutional Neural Networks, 2. Review 1 Preparation |
| Day 11 | 19/11/2021 | Data Augmentation, Regularization & ResNets |
| Day 12 | 20/11/2021 | Data Augmentation, Regularization & ResNets |
| Day 13 | 22/11/2021 | Image transformations, flip, exposure, Filters, Gaussian Filter, Histogram |
| Day 14 | 23/11/2021 | Custom Filters, Morphology Operations, Contrast and Brightness |
| Day 15 | 24/11/2021 | Image transformations, flip, exposure, Filters, Gaussian Filter, Histogram, Custom Filters, Morphology Operations, Contrast and Brightness. |
| Day 16 | 25/11/2021 | Residual Networks (ResNets) : Sigmoid Activation Function, Optimizer, Forward and Backward Propagation in Gradient Descent method, Vanishing Gradient problem. |

|  |  |  |
| --- | --- | --- |
| **Day** | **Date** | **Task Done** |
| Day 17 | 26/11/2021 | ResNet50, Data Augmentation : Under Sampling, Over Sampling by duplication, SMOTE, Ensemble method, Focal Loss. |
| Day 18 | 27/11/2021 | Data Augmentation on TB negative image, Implementing ResNet50 on TB dataset. |
| Day 19 | 29/11/2021 | 1.ResNet50 : Model Training, Evaluation and plot. 2.Weekely Meeting. |
| Day 20 | 30/11/2021 | Understanding ResNet50 coding |
| Day 21 | 01/12/2021 | Understanding ResNet50 coding |
| Day 22 | 02/12/2021 | Inception V3 Layer Architecture : Inception A, Inception C, Inception B, Reduction A, Reduction B, Auxilary Classifier |
| Day 23 | 03/12/2021 | Practical Implementation of Inception V3. |
| Day 24 | 04/12/2021 | Practical Implementation of Inception V3 : Solving Error |
| Day 25 | 06/12/2021 | Practical Implementation of Inception V3 : Sloved the Error |
| Day 26 | 07/12/2021 | Inception V3 another method |
| Day 27 | 08/12/2021 | 1. Solved the Error, 2. Inception V3 on TB dataset |
| Day 28 | 09/12/2021 | Refering VGG16 |
| Day 29 | 10/12/2021 | Refering VGG16 |
| Day 30 | 11/12/2021 | Classificarion team meet |
| Day 31 | 13/12/2021 | 1. Learning about Localization 2. Understanding BatchNormalizaiton and Dilution |
| Day 32 | 14/12/2021 | Read XML file to create Bounding Boxes. |
| Day 33 | 15/12/2021 | Bounding Boxes. Create Mask |
| **Day** | **Date** | **Task Done** |
| Day 34 | 16/12/2021 | Create Mask |
| Day 35 | 17/12/2021 | Instructions regarding Paper writing meeting. |
| Day 36 | 18/12/2021 | Modified Augmentation code to get better TB negative images |
| Day 37 | 20/12/2021 | 1. Worked on ResNet50 and InceptionV3 to get better accuracy. 2. Preparing PPT. |
| Day 38 | 21/12/2021 | Review 2 |
| Day 39 | 22/12/2021 | Completed Abstract |
| Day 40 | 23/12/2021 | Related Works, Method - InceptionV3 Architecture, Experimental results and discussion |
| Day 41 | 24/12/2021 | 1. Related Works, 2.Method - about InceptionV3 and ResNet50 |
| Day 42 | 25/12/2021 | Understanding ResNet50 coding |
| Day 43 | 26/12/2021 | Understanding ResNet50 coding |
| Day 44 | 27/12/2021 | Prepare for final review |
| Day 45 | 28/12/2021 | Prepare for final review |
| Day 46 | 29/11/2021 | Final review |